

Sistemas de Tempo-Real

Notas de curso realizado em Agosto de 2006 na
Universidade Federal do Rio Grande do Norte, Natal, Brasil



Universidade do Porto
Faculdade de Engenharia



Francisco Vasques
Faculdade de Engenharia da
Universidade do Porto
<http://www.fe.up.pt/~vasques>

1. Introdução aos Sistemas de Tempo-Real



Universidade do Porto
Faculdade de Engenharia

Plano das Aulas

- **Introdução aos Sistemas de Tempo-Real**
 - Motivação para o estudo de STR
 - Definições
 - Concepções Erradas
 - Classificação
 - Pontos Chave para a Conceção de STR
 - Desafios de Investigação
- **Escalonamento de Tempo-Real**
- **Protocolos de Comunicação de Tempo-Real**

Motivação para o estudo de STR

■ Sistemas de Tempo-Real (STR)

- Sistemas caracterizados por estarem sujeitos a *requisitos associados à passagem do tempo*.

■ Âmbito dos STR

- ... desde controladores de pequenos electrodomésticos, até sistemas de controlo de tráfego aéreo ou ferroviário...
- ... desde controladores robóticos até aplicações de vídeo-conferência...

■ Aserção fundamental

- A utilização de sistemas computacionais para o controlo de actividades com requisitos temporais, requer a *selecção / utilização de metodologias adequadas* para o seu correcto desenvolvimento.

Motivação para o estudo de STR

■ Práticas correntes em STR (insuficientes)

- Considerar como suficiente a utilização de um *Sistema Operativo de Tempo-Real*, que suporte interrupções, concorrência e mecanismos de temporização com a granularidade adequada;
- Determinar o *nível de prioridades de tarefas* a partir da sua importância relativa (em termos de funcionalidade), ignorando as restrições temporais associadas à sua computação
 - » Este é um dos *principais problemas* associados às práticas correntes na concepção de Sistemas de Tempo-Real.

Motivação para o estudo de STR

■ Práticas correntes em STR (insuficientes)

- Caso não seja utilizado um Sistema Operativo de Tempo-Real, a utilização de *linguagens de baixo nível* para efectuar temporizações particulares aumenta a complexidade global da aplicação e dificulta a integração dos seus diferentes componentes.

■ Consequência

- A utilização de *metodologias e ferramentas convencionais* para o desenvolvimento de sistemas de tempo-real pode ter como consequência *comportamento temporalmente imprevisível* das aplicações desenvolvidas, não oferecendo uma adequada garantia de correcto funcionamento.

Motivação para o estudo de STR

■ Áreas de aplicação dos STR:

- » Diferentes níveis de requisitos em termos de disponibilidade, fiabilidade, manutenibilidade, segurança e/ou integridade de dados
- Telecomunicações/Comunicações de dados: Comutação telefónica; Vídeo-conferência; CSCW; Aplicações Multimédia;
- Aeroespacial: Aviónica; Satélites;
- “Drive-By-Wire”: Controlo de motor, ABS, segurança, conforto, etc.
- Defesa: Controlo de mísseis; Radar; Sonar;
- Automação Industrial: Controlo de processos / manufactura;
- Sector Financeiro: Transacções em bolsa; Sistemas bancários on-line;
- Transportes: Sinalização ferroviária;



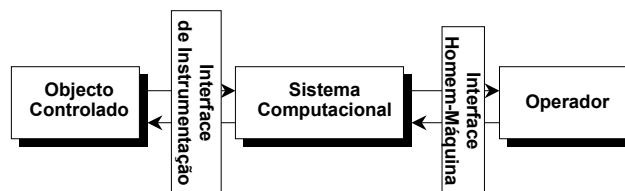
Plano das Aulas

■ Introdução aos Sistemas de Tempo-Real

- Motivação para o estudo de STR
- Definições
- Concepções Erradas
- Classificação
- Pontos Chave para a Concepção de STR
- Desafios de Investigação



Definições

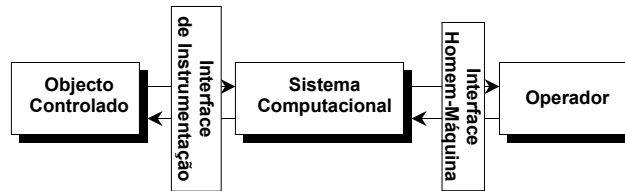


■ Sistema de Tempo-Real

- Um Sistema de Tempo-Real é um *sistema computacional reactivo*, i.e., que reage a estímulos externos (incluindo à passagem do tempo) em intervalos de tempo impostos pelo seu ambiente (operador e objecto controlado);
- Um Sistema de Tempo-Real pode ser caracterizado por ter, em geral, um *funcionamento contínuo* que lhe permite reagir a estímulos externos.



Definições

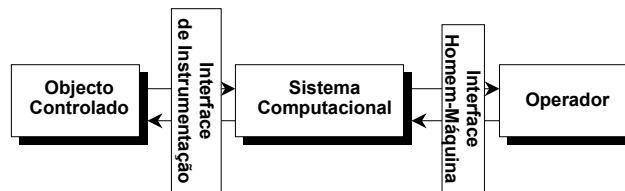


■ Sistema de Tempo-Real

- A *correção de um sistema de tempo-real* depende não só do resultado lógico das computações efectuadas, mas também do instante de tempo em que os resultados são produzidos [Stankovic, 1988].
- Um Sistema de Tempo-Real não é um sistema rápido, mas sim um sistema previsível.



Definições



■ Sistemas genéricos vs. Sistemas de Tempo-Real

“Executar uma determinada tarefa no menor intervalo de tempo possível”

→ objectivo de um “*general purpose system*”

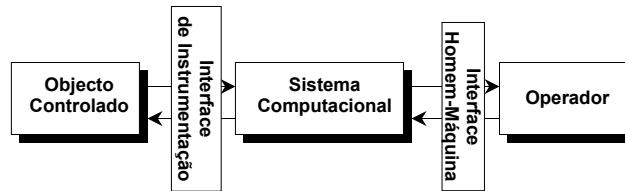
vs.

“Nunca ultrapassar o intervalo de tempo pré-determinado para executar uma determinada tarefa”

→ objectivo de um STR (a consequência da perda de uma meta temporal pode ser drástica)



Definições

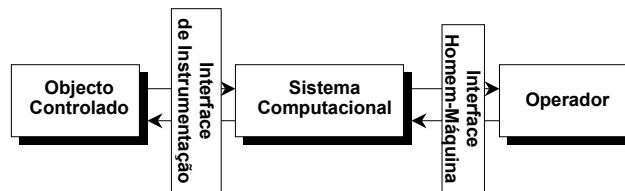


■ Sistemas Embebido (“*Embedded System*”)

- É um sistema no qual a parte “*física*” e a parte computacional estão fortemente integradas, interagindo através de sensores e actuadores para desempenhar uma *função específica*;
- Um Sistema Embebido é também um *sistema computacional reactivo*, i.e., que reage a estímulos externos em intervalos de tempo impostos pelo seu ambiente;
- Exemplos:
 - » “Pacemaker” e desfibrilhador;
 - » Leitor de cartões inteligentes;
 - » Receptor GPS.



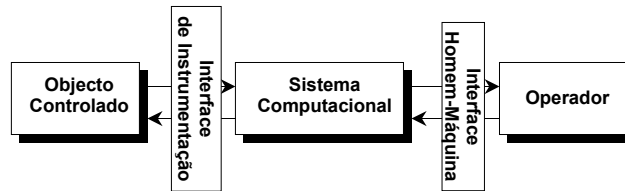
Definições



■ Sistemas Embebido (“*Embedded System*”)

- » Um sistema embebido é (quase) sempre um sistema de tempo-real;
 - Contra exemplo: Um PDA que suporte uma aplicação lúdica.
- » Nem todos os sistemas de tempo-real são sistemas embebidos:
 - Sistemas de controlo de tráfego aéreo;
 - Sistemas de negociação bolsista.

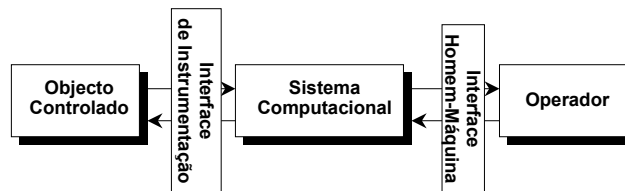
Definições



■ Parâmetros Temporais

- Os *requisitos temporais* de um sistema de tempo-real são impostos pelo seu ambiente, assim como a execução das suas aplicações é também frequentemente controlada pelo ambiente:
 - » impossibilidade do sistema computacional controlar a cadência e/ou a duração dos estímulos provenientes do seu ambiente;
 - » tempos de resposta requeridos ao sistema computacional serem impostos pelo ambiente.

Definições



■ Parâmetros Temporais

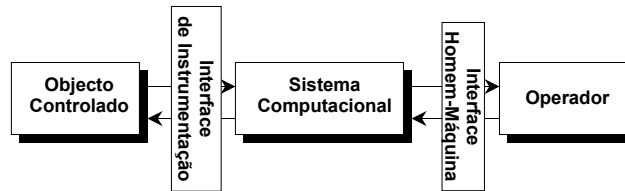
- Exemplo de *requisitos temporais* impostos pelo ambiente de um sistema de tempo-real:
 - » *Meta temporal* para finalização de uma tarefa (melhor caso e pior caso)



- A ultrapassagem de uma meta temporal ("*deadline*") corresponde a uma *avaria temporal* (quando o serviço prestado não está em conformidade com a especificação).



Definições



■ Parâmetros Temporais

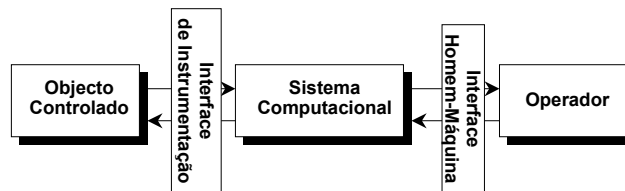
– Exemplo de *requisitos temporais* impostos pelo ambiente de um sistema de tempo-real:

» Sincronização entre instantes de produção de resultados:

■ Ex. Sincronização Som – Imagem



Definições

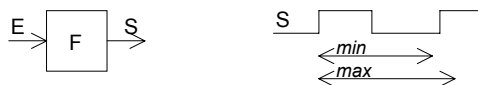


■ Parâmetros Temporais

– Exemplo de *requisitos temporais* impostos pelo ambiente de um sistema de tempo-real:

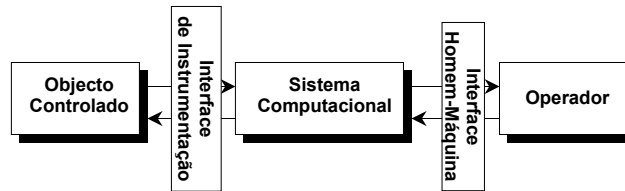
» Cadência de produção:

■ Ex. Regularidade de apresentação de imagens num vídeo





Definições



■ Parâmetros Temporais

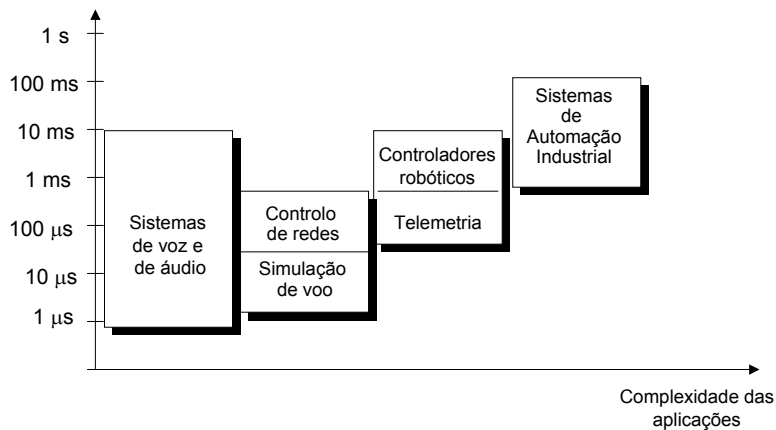
- *Tempo de Resposta*: Intervalo entre o instante em que uma ou mais entradas modificam o seu valor e o instante em que o sistema computacional reage a essas modificações, através de uma modificação do valor das suas saídas.



Definições

■ Parâmetros Temporais

- Tempo de Resposta



■ Introdução aos Sistemas de Tempo-Real

- Motivação para o estudo de STR
- Definições
- Conceções Erradas
- Classificação
- Pontos Chave para a Conceção de STR
- Desafios de Investigação

Concepções Erradas sobre STR

■ Tempo-real significa execução rápida (errado!)

- “*It does not matter so much whether wrong scheduling decisions or, at best poor scheduling decisions, are made quickly or not*”, G. Le Lann (1990)
- Os requisitos temporais podem ser críticos e não ser curtos (ex. controlo de temperatura de uma caldeira).

■ Caso um processador não preencha os requisitos de um STR, então bastará utilizar um processador mais rápido (errado!)

- A potência de cálculo é uma condição necessária mas não suficiente para a garantia dos requisitos temporais (conforme se verá ao longo desta disciplina).

Concepções Erradas sobre STR

- **Não é necessária fundamentação científica para a concepção / desenvolvimento de Sistemas de Tempo-Real (errado!)**
 - Poderia ser verdade há 25 anos (!), mas quanto maior é a complexidade dos sistemas, maior a necessidade de uma forte fundamentação teórica para a sua concepção / desenvolvimento;
 - » Escalonamento de tempo-real; verificação formal; etc.

Concepções Erradas sobre STR

- **O ambiente de um STR é tipicamente estático (errado!)**
 - Existe um cada vez maior número de aplicações de tempo-real com ambientes dinâmicos...
- **A utilização de um computador para suporte de um STR implica a programação de *device drivers* (errado!)**
 - Existem já linguagens de alto nível que integram “gestão de políticas de escalonamento”, como por exemplo Ada05;
 - Existem Sistemas Operativos de Tempo-Real que...



Concepções Erradas sobre STR

■ Não vale a pena discutir como garantir o desempenho de tempo-real de um sistema devido às imperfeições do:

- » *software* (que sempre terá “bugs”);
- » *hardware* (que sempre terá falhas)
- » *ambiente de operação* (para o qual nunca se conhecerão *a priori* as *reais* condições de operação)

(errado!) ... conforme se verá ao longo da disciplina!

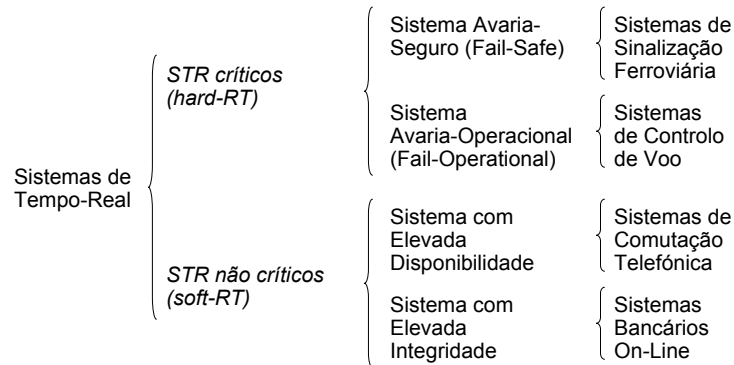


Plano das Aulas

■ Introdução aos Sistemas de Tempo-Real

- Motivação para o estudo de STR
- Definições
- Concepções Erradas
- Classificação
- Pontos Chave para a Conceção de STR
- Desafios de Investigação

Classificação dos Sistemas de Tempo-Real

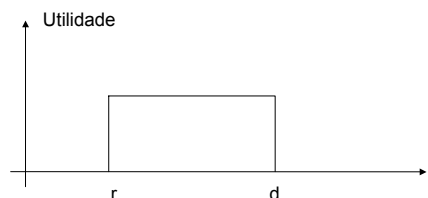


Classificação dos Sistemas de Tempo-Real

■ Sistema Crítico vs. Sistema Não-Crítico

- Considera-se um STR como sendo *crítico* caso o não cumprimento de uma meta temporal ("deadline") tenha *consequências graves* para a utilidade do sistema (prejuízo causado várias ordens de grandeza superior ao benefício decorrente da sua correcta execução);

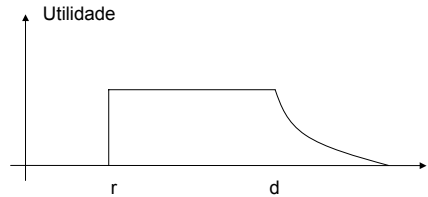
» Necessidade de *obtenção de garantias* em fase de concepção;



Classificação dos Sistemas de Tempo-Real

■ Sistema Crítico vs. Sistema Não-Crítico

- Considera-se um STR como sendo *não-crítico* caso esse não cumprimento de metas temporais tenha unicamente *consequências ligeiras* para a utilidade do sistema;
- Enquanto que num sistema crítico as metas temporais são de cumprimento “obrigatório”, num sistema não-crítico as metas temporais descrevem o comportamento temporal desejado para o sistema (por exemplo numa aplicação de vídeo-conferência)

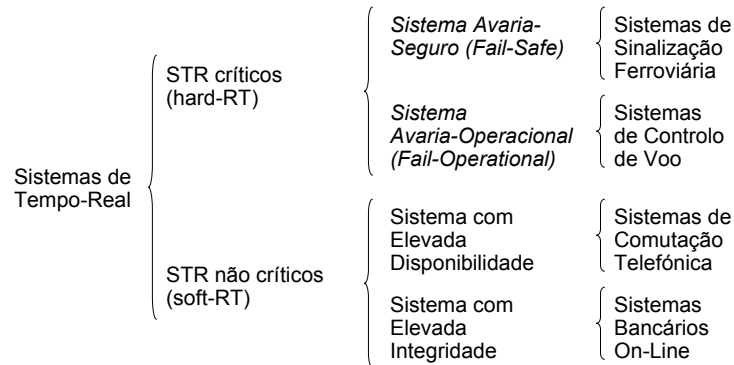


Classificação dos Sistemas de Tempo-Real

■ Num Sistema de Tempo-Real não-crítico são aceitáveis as seguintes ocorrências:

- » uma meta temporal (“*deadline*”) ocasionalmente ultrapassada (estando definido o número máximo de ultrapassagens por intervalo de tempo);
- » a prestação de um serviço pode estar ocasionalmente atrasada (estando definido o atraso máximo aceitável).
- As *aplicações interactivas* podem ser consideradas, na generalidade dos casos, aplicações não-críticas em termos temporais.
 - » ... apesar de tipicamente serem apresentadas como exemplo de aplicações de tempo-real!

Classificação dos Sistemas de Tempo-Real

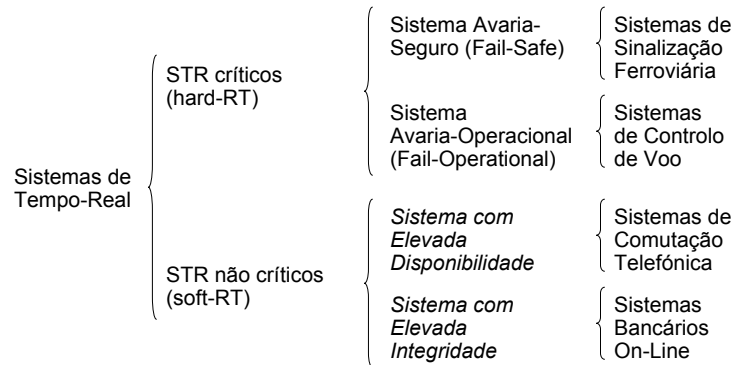


Classificação dos Sistemas de Tempo-Real

■ Sistema Avaria-Seguro vs. Sistema Avaria-Operacional

- Caso *existam um ou mais estados seguros* de funcionamento do objecto controlado, em caso de avaria o sistema computacional deve evoluir para um desses estados ("Fail-Safe");
- Caso *não existam estados seguros* de funcionamento, o sistema deve manter um nível mínimo de funcionalidade com segurança ("Fail-Operational System");
- *Nota:* um estado seguro de funcionamento é função do objecto controlado e não do sistema computacional.

Classificação dos Sistemas de Tempo-Real



Classificação dos Sistemas de Tempo-Real

■ Sistemas de Tempo-Real Não-Críticos

- Os STR não críticos podem ser classificados em função de parâmetros que influenciem decisivamente a *utilidade do sistema* (parâmetros que, em caso de incumprimento, provocam um prejuízo várias ordens de grandeza superior ao benefício decorrente da correcta execução do STR). Por exemplo:
 - » Elevada *disponibilidade* de serviço, no caso de um sistema de comutação telefónica;
 - » Elevada *integridade* na manipulação de dados, no caso de um sistema bancário interactivo.

Classificação dos Sistemas de Tempo-Real

■ Classificação Ortogonal:

Sistemas de Resposta Garantida vs. Sistemas de Melhor Esforço

- Considera-se um sistema como sendo de *Resposta Garantida* (“Guaranteed Response”) caso a sua concepção seja baseada na *adequabilidade de recursos*, ou seja, na garantia da existência de recursos computacionais suficientes para suportar os cenários máximos de carga e de falhas;
- Sistema *determinístico*, no que diz respeito à previsibilidade temporal;
- A *probabilidade de avaria* de um sistema de resposta garantida é igual à “*não cobertura dos pressupostos*” (ou seja, igual à probabilidade de os pressupostos de carga e de falhas não estarem correctos).

Classificação dos Sistemas de Tempo-Real

■ Classificação Ortogonal:

Sistemas de Resposta Garantida vs. Sistemas de Melhor Esforço

- Considera-se um sistema como sendo de *Melhor Esforço* caso a sua concepção seja baseada em estratégias de *alocação dinâmica de recursos*, combinadas com argumentos probabilísticos acerca da simultaneidade na ocorrência de cenários máximos de carga e/ou de falhas;
- Sistema *probabilístico* no que diz respeito à previsibilidade temporal .

Classificação dos Sistemas de Tempo-Real

■ Vantagens / Desvantagens de um Sistema de Resposta Garantida

- O cumprimento das metas temporais é garantido na fase de concepção, não havendo a necessidade de sobrecarregar o processamento “on-line” com o cálculo de testes de escalonabilidade;
- Vantajoso para aplicações de elevada criticidade, devido ao determinismo inerente;
- Requer na fase de concepção um conhecimento exacto dos pressupostos de carga e de falhas, o que poderá ser de difícil exequibilidade;
- Implica uma alocação de recursos para o pior caso (carga máxima), gerando uma grande sub-utilização de recursos computacionais;

Plano das Aulas

■ Introdução aos Sistemas de Tempo-Real

- Motivação para o estudo de STR
- Definições
- Concepções Erradas
- Classificação
- Pontos Chave para a Concepção de STR
- Desafios de Investigação

Pontos chave para a concepção de aplicações

■ **Determinismo de execução: Previsibilidade**

- Deverão ser conhecidos todos os cálculos efectuados para a *determinação da carga* imposta sobre o sistema computacional (aplicação e sistema operativo);
- Deverão ser conhecidos os *tempos de execução* das aplicações suportadas (no pior caso, e de uma forma não subestimada)
 - » Só será possível caso o hardware utilizado tenha tempos de execução previsíveis.

Pontos chave para a concepção de aplicações

■ **Determinismo de execução: Previsibilidade**

- A correcta selecção da *ordem de execução* das tarefas/funções é uma das funções mais relevantes num STR:
 - » Esta selecção deverá ser efectuada através da utilização de *algoritmos de escalonamento adequados*, para os quais existam provas matemáticas da sua correcta execução.
- Deverá ser obtida uma *prova de correcto funcionamento do STR* (garantia dos requisitos temporais) em tempo de concepção, a partir dos pressupostos de carga e de falhas.
 - » A utilização de actividades de *teste* não é suficiente, pois estas não garantem a verificação em casos extremos de funcionamento.

Pontos chave para a concepção de aplicações

■ Limite à previsibilidade de um STR

- Na fase de concepção de um STR, a sua previsibilidade depende de um conjunto de pressupostos relativos ao ambiente:
 - » *Pressupostos de Carga*: definição da carga máxima do sistema computacional, através da especificação da taxa máxima de transacções entre o sistema computacional e o ambiente;
 - » *Pressupostos de Falhas*: definição do tipo e frequência de falhas que o sistema deve suportar (tolerar);
- É fundamental que os pressupostos relativos ao ambiente estejam de acordo com a realidade de funcionamento (Cobertura de Pressupostos).

Plano das Aulas

■ Introdução aos Sistemas de Tempo-Real

- Motivação para o estudo de STR
- Definições
- Concepções Erradas
- Classificação
- Pontos Chave para a Concepção de STR
- Desafios de Investigação

Desafios de Investigação

■ Áreas de Investigação em STR:

- Especificação e Verificação Formal
 - » Incorporação da variável *tempo* na especificação/verificação de sistemas;
 - » Investigação na área de fronteira entre os sistemas síncronos e os sistemas assíncronos;
 - » Necessária análise quantitativa (perda de metas temporais, taxas de avarias, etc.) e não só qualitativa (a que é tradicionalmente efectuada na área da verificação);
 - » Necessárias metodologias eficazes para a resolução do problema de explosão de estados.
- Análise de Desempenho
 - » Baseada unicamente em simulação e teste. Necessária análise formal para sistemas dinâmicos e ambientes imprevisíveis.

Desafios de Investigação

■ Áreas de Investigação em STR:

- Sistemas Operativos de Tempo-Real
 - » Problema da multiplicidade (e opacidade) de opções:
 - Para sistemas críticos: VxWorks;
 - Para sistemas não-críticos: Windows CE;
 - Muitos outros para diferentes domínios de aplicação.
- Linguagens de Programação de Tempo-Real
 - » Suporte à gestão do tempo;
 - » Suporte à análise de escalonabilidade;
 - » Suporte à modularidade e à re-utilização de software;
 - » Suporte para programação distribuída;
 - » Suporte para tolerância a falhas.

Desafios de Investigação

■ Áreas de Investigação em STR:

- Engenharia de Software
 - » Tradicionalmente, só os aspectos funcionais são endereçados;
 - » Necessário também considerar:
 - Tempo, Confiança no Funcionamento, QoS
 - Evolutibilidade;
 - Modularidade;
 - Adaptabilidade e configurabilidade.

Desafios de Investigação

■ Áreas de Investigação em STR:

- Bases de Dados Distribuídas de Tempo-Real
 - » Necessidade de relaxamento de critérios de serialização global;
 - » Controlo de concorrência de tempo-real.
- Tolerância a Falhas
 - » O tratamento de erros/falhas deve considerar as restrições temporais das aplicações.
- Teoria de Escalonamento de Tempo-Real
 - » Tema já bem estudado!
- Arquitectura de Computadores para Sistemas de Tempo-Real
 - » Sistemas *cada vez mais* distribuídos;
 - » Arquitecturas tolerantes a falhas.

■ Áreas de Investigação em STR:

- Sistemas Multimedia de Tempo-Real
 - » Necessária uma especificação de requisitos (previsibilidade) adequada;
 - » Focalização em aspectos relacionados com a garantia de QoS e o escalonamento das comunicações.
- Protocolos de Comunicação de Tempo-Real
 - » Escalonamento de comunicações integrado com o escalonamento de tarefas nos nós, por forma a fornecer soluções de escalonamento global;
 - » Soluções para roteamento dinâmico de tempo-real;
 - » Comunicações tolerantes a falhas e seguras.